

## چکیده:

رجیستری ویندوز یک پایگاه داده، حاوی اطلاعات پیکربندی سامانه عامل ویندوز و اطلاعات تنظیمات سخت افزار و نرم افزار است. در این مقاله چگونگی دسترسی به این پایگاه داده توسط برنامه (C-Sharp) بررسی می شود.

در پنجمین ویراست از کتاب «فرهنگ رایانه‌ای مایکروسافت» درباره رجیستری چنین آمده است: یک پایگاه داده متمرکز با ساختار سلسله مراتبی که در ویندوزهای CE، NT، 9x و 2000 به کار رفته و از آن برای نگهداری اطلاعات لازم جهت پیکربندی سیستم برای یک یا چند کاربر، برنامه‌های کاربردی و تجهیزات سخت‌افزاری استفاده می‌شود. رجیستری، حاوی اطلاعاتی است که ویندوز در حین عمل، به طور پیوسته و مکرر به آنها رجوع می‌کند؛ از قبیل [تنظیمات مربوط به] پرونده‌های هر کاربر، برنامه‌های کاربردی نصب شده بر روی رایانه، انواع مستنداتی که [هر کاربر] می‌تواند ایجاد کند، خصلت‌های صفحه‌ای پوشه‌ها، آیکون‌های برنامه‌ها، سخت‌افزارهای موجود در سیستم، درگاه‌های مورد استفاده و... ( ماخذ: ویکی‌پدیا فارسی)

می توانید در رجیستری اطلاعات پیکربندی برنامه را ذخیره کنید به طور مثال آدرس پردازشگر، رشته اتصال به پایگاه داده، تنظیمات کاربر و یا حتی آخرین کاربری که وارد برنامه شده در این صورت هرگاه برنامه اجرا شود نام آخرین کاربری که از سامانه استفاده کرده در فیلد نام کاربری ظاهر شده و کاربر نیازی به وارد کردن آن نخواهد داشت.





نکاتی که باید در بکارگیری رجیستری مورد توجه قرار گیرد:

۱- از کلید CurrentUser استفاده کنید زیرا استفاده از LocalMachine باعث می شود برنامه شد فقط در Privileged mode اجرا شود یعنی باید کاربر عضو گروه مدیر سیستم باشد.

۲- اگر ضرورت ندارد کلید های خود را پراکنده نکنید. آنها را زیر یک کلید اصلی قرار دهید. پراکنده کردن کلیدها نصب زدایی uninstall را دشوار میکند.

۳- اگر به هر دلیل رجیستری تغییر کرده و برنامه قادر به خواندن اطلاعات نبود، نباید با خطای زمان اجرا مواجه شویم بهتر است تنظیمات پیش فرض از یک فایل XML خوانده شده و رجیستری بازسازی گردد.

۴- اگر می خواهید اطلاعات حساس و محرمانه مانند رمزهای عبور، تاریخ انقضای برنامه، رشته اتصال به پایگاه داده را در رجیستری ذخیره نمایید آن را رمز Encrypt نمایید زیرا رجیستری ویندوز برای کاربران براحتی قابل دسترسی است.

۵- برای نامگذاری کلید ها بهتر است از نام هایی نظیر k001 استفاده کنید، استفاده از نامهایی که بیان کننده محتوای کلید هستند مانند password or bkcolor مطلوب نیست زیرا کاربر متوجه کاربرد آن کلیدها شده با استفاده از ابزارهای ویرایش رجیستری مقادیر آنها را تغییر می دهد.



طرح مسئله:

می خواهیم یک برنامه بنویسم که در آن تنظیمات کاربر در رجیستری ذخیره شود و هرگاه برنامه اجرا شود این تنظیمات خوانده شده و بکار گرفته شود. برای آنکه بهتر برنامه را بشناسید آنرا از این لینک دانلود کنید و کد برنامه را مطالعه کنید:



■ در برنامه بسیار ساده ما دو کلاس و دو فرم وجود دارند:

فرم اصلی برنامه و فرم تنظیمات کاربر دو کلاس هم استفاده شده یک کلاس برای دسترسی به رجیستری با امکان بازسازی رجیستری خراب. توابع این کلاس دارای خروجی های مناسبی هستند که در صورت وقوع خطا/استثنا وضعیت را بدرستی نشان میدهند.

اصل برنامه شامل خواندن از رجیستری، نوشتن در آن و حذف و ایجاد کلید جدید است:

■ خواندن از رجیستری:

```
public int ReadConfig(string KeyName, out string StrOut)
{
```

```
RegistryKey pRegKey = Registry.CurrentUser;

string Str1;

StrOut = "";

try
{
    pRegKey = pRegKey.OpenSubKey(@"SOFTWARE\D0020\L01");

    if (pRegKey == null)
    {
        return 1;
    }
    else
    {
        if (pRegKey.GetValue(KeyName) != null)
        {
            Str1 = pRegKey.GetValue(KeyName).ToString();

            pRegKey.Close();

            if (Str1 == "")
            {
```

```
        return 2;
    }
    else
    {
        StrOut = Str1;
        return 0;
    }
}
else
{
    return 3;
}
}
}
catch
{
    return 4;
}
}
```

```
public int WriteConfig(string KeyName, string StrIn)
{
    RegistryKey pRegKey = Registry.CurrentUser;

    try
    {
        pRegKey = pRegKey.OpenSubKey(@"SOFTWARE\D0020\L01", true);

        if (pRegKey == null)
        {
            return 1;
        }
        else
        {
            pRegKey.SetValue(KeyName, StrIn);
            pRegKey.Close();
            return 0;
        }
    }
}
```

```
catch
{
    return 2;
}
}
```

اصلاح و تغییر (افزودن کلید حذف کلید)

```
public int RepairRegistry()
{
    try
    {
        RegistryKey pRegKey1 = Registry.CurrentUser;
        pRegKey1 = pRegKey1.OpenSubKey(@"SOFTWARE", true);

        if (pRegKey1 != null)
        {
            pRegKey1.DeleteSubKeyTree("D0020"); // پاک کردن
            pRegKey1.Close();
        }
    }
}
```

```
catch
{
    // do nothing
}
try
{
    RegistryKey pRegKey1 = Registry.CurrentUser;
    pRegKey1 = pRegKey1.OpenSubKey(@"SOFTWARE", true);

    if (pRegKey1 == null)
    {
        pRegKey1.Close();
        return 1;
    }
    else
    {
        string StrConf = GetConfigFromXml("MainConfig", 0, 1);
        if (StrConf != "ERROR READING CONF FILE")
        {
            pRegKey1.CreateSubKey(@"D0020\L01");//ایجاد کلید
```

```
        WriteConfig("D002001", StrConf);

        pRegKey1.Close();

        return 0;
    }

    else
    {
        return 2;
    }
}

catch
{
    return 3;
}
}
```

زسازی رجیستری:

نطور که در ابتدای بحث مطرح شد بهتر است برنامه را طوری طراحی کنید که در صورت خراب شدن رجیستری بتواند تنظیمات خود را از یک فایل دریافت کند. اگر

برنامه را اجرا کنید سپس با REGEDIT کلید D0020 را حذف کنید در اجرای مجدد برنامه با خطا و سپس با پیام زیر روبرو میشوید. برنامه رجیستری را اصلاح میکند.



■ رمز کردن اطلاعات:

برای رمزگذاری الگوریتم های متعددی وجود دارد. در دات نت کلاس ویژه رمزگذاری با نام **System.Security.Cryptography** وجود دارد. اما در این برنامه از یک الگوریتم بسیار ساده برای رمز کردن یک عدد بین 0 تا 3 استفاده شده:

دو عدد دو رقمی تصادف ایجاد میشوند قدرمطلق تفاضل آن دو عدد با عدد ورودی جمع شده و هر سه عدد به ترتیب پشت سرهم قرار گرفته و یک رشته پنج یا شش کاراکتری را میسازند. برای رمزگشایی نیز الگوریتم معکوس اجرا میشود:

```
public static string EncodeConf(int i)
{
    Random myrandom = new Random(System.DateTime.Now.Millisecond);

    int i1, i2, d;

    i1 = myrandom.Next(10, 99);
```

```
        i2 = myrandom.Next(10, 99);

        d = i + System.Math.Abs(i1 - i2);

        string StrOut = i1.ToString() + i2.ToString() + d.ToString();

        return StrOut;

    }

    public static int DecodeConf(string StrIn)
    {

        string Str1, Str2, Strd;

        Str1 = StrIn.Substring(0, 2);

        Str2 = StrIn.Substring(2, 2);

        Strd = StrIn.Substring(4, StrIn.Length - 4);

        int i, i1, i2, d;

        i1 = Int32.Parse(Str1);

        i2 = Int32.Parse(Str2);

        d = Int32.Parse(Strd);

        i = d - System.Math.Abs(i1 - i2);
    }
}
```

```
return i;
```

```
}
```